

Sistemas Gerenciadores de Banco de Dados: Transações e Controle de Concorrência

João Eduardo Ferreira e Helio Navarro

**Elmasri, R.; Navathe, S. B. Sistemas de Banco de Dados, 4ed.
Pearson Addison Wesley, 2005 (Caps. 17 e 18)**

**Silberschatz, A.; Korth, H. F., Sudarshan, S. Sistema de Bancos de
Dados. 3a. Edição, Makron Books, 1998. 778p. (Caps 13 e 14)**

Tópicos

(1) SGBD

(2) Sistemas Monousuários e Multiusuários

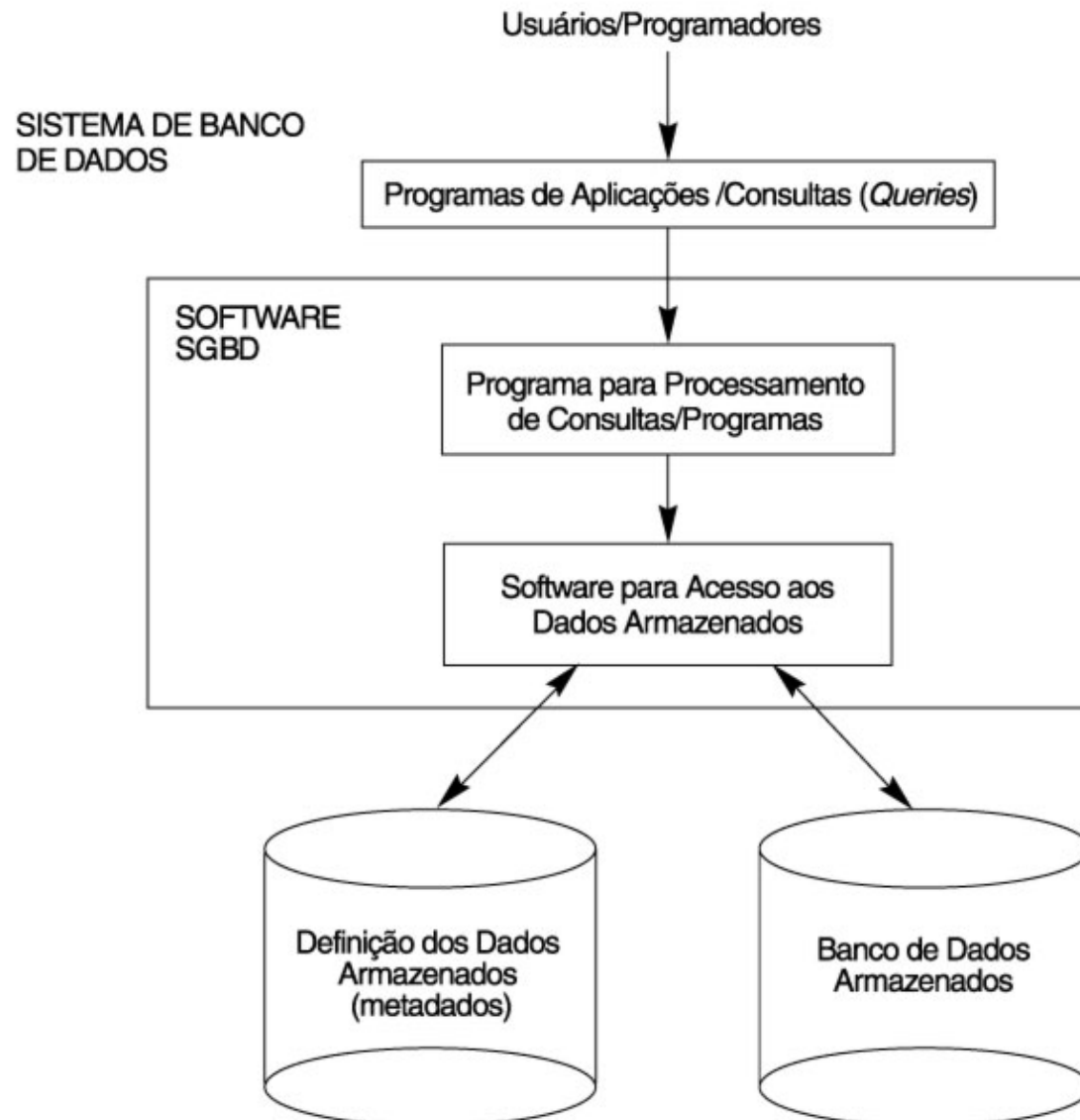
(3) Definição e Estrutura da Transação

(4) Propriedades da Transação

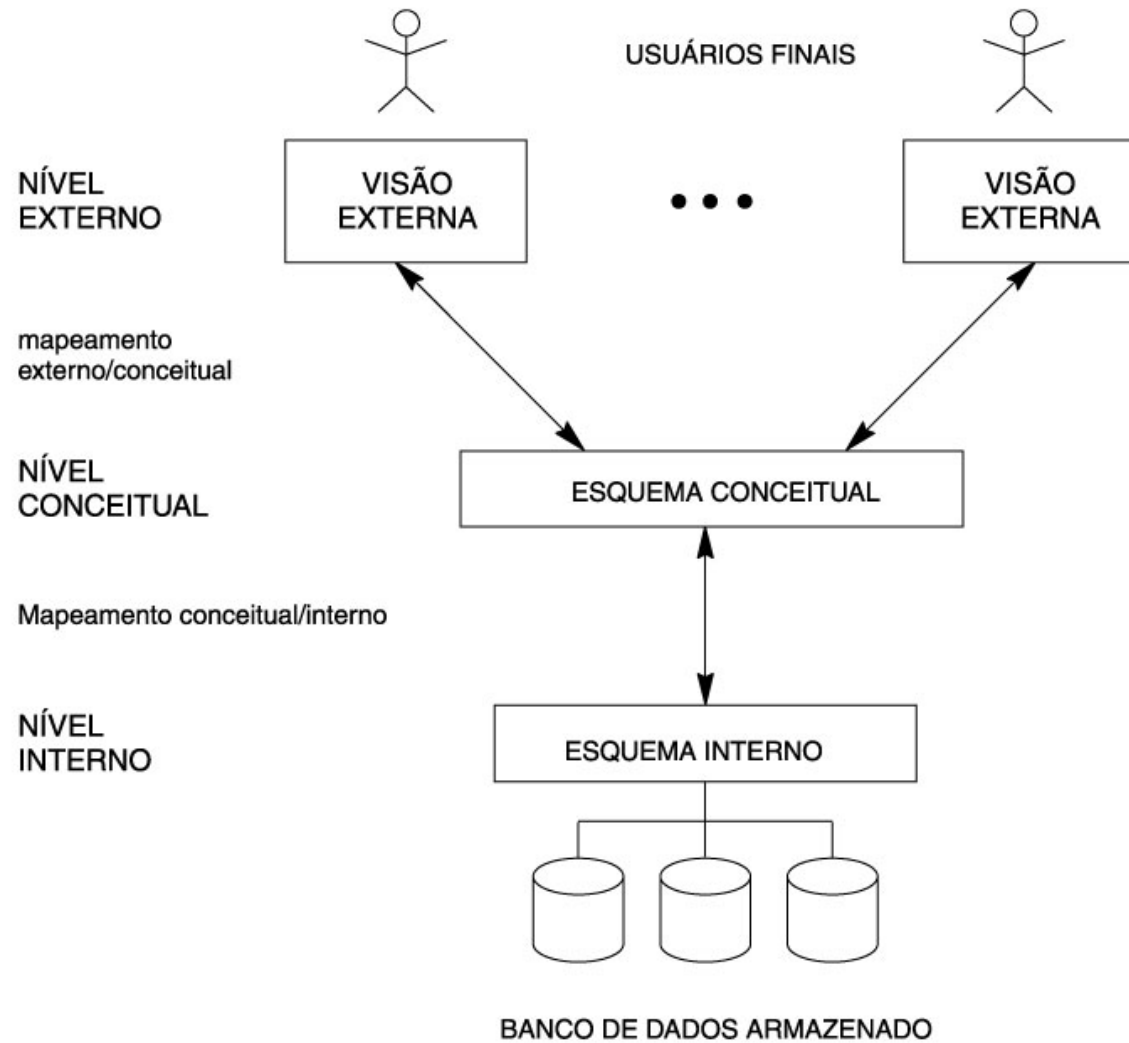
(5) Planos de Execução das Transações

(6) Controle de Concorrência

Sistema de Banco de Dados



Arquitetura do SBD: Três esquemas



BD – Informações sobre uma Empresa

EMPREGADO	PNOME	MINICIAL	UNOME	SSN	DATANASC	ENDERECO	SEXO	SALARIO	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT	LOCALIZACOES	DNUMERO	DLOCALIZACAO
		1	Houston
		4	Stafford
		5	Bellaire
		5	Sugarland
			Houston

DEPARTAMENTO	DNOME	DNUMERO	GERSSN	GERDATAINICIO
	Pesquisa	5	333445555	1988-05-22
	Administração	4	987654321	1995-01-01
	Sede administrativa	1	888665555	1981-06-19

TRABALHA_EM	ESSN	PNO	HORAS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

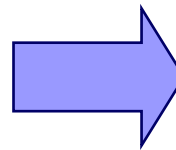
PROJETO	PJNOME	PNUMERO	PLOCALIZACAO	DNUM
	ProdutoX	1	Bellaire	5
	ProdutoY	2	Sugarland	5
	ProdutoZ	3	Houston	5
	Automatização	10	Stafford	4
	Reorganização	20	Houston	1
	Novos Benefícios	30	Stafford	4

DEPENDENTE	ESSN	NOME_DEPENDENTE	SEXO	DATANASC	PARENTESCO
	333445555	Alice	F	1986-04-05	FILHA
	333445555	Theodore	M	1983-10-25	FILHO
	333445555	Joy	F	1958-05-03	CÔNJUGE
	987654321	Abner	M	1942-02-28	CÔNJUGE
	123456789	Michael	M	1988-01-04	FILHO
	123456789	Alice	F	1988-12-30	FILHA
	123456789	Elizabeth	F	1967-05-05	CÔNJUGE

Sistemas monousuários e multiusuários

SGBD Monousuário: somente um usuário de cada vez pode acessar o SBD

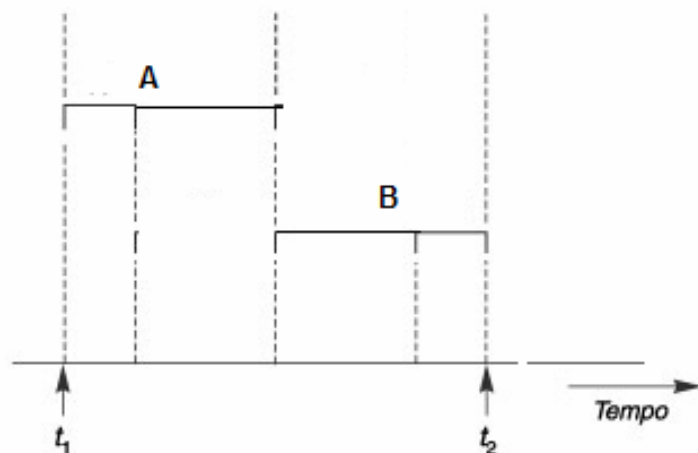
SGBD Multiusuário:
muitos usuários podem
acessar o SBD



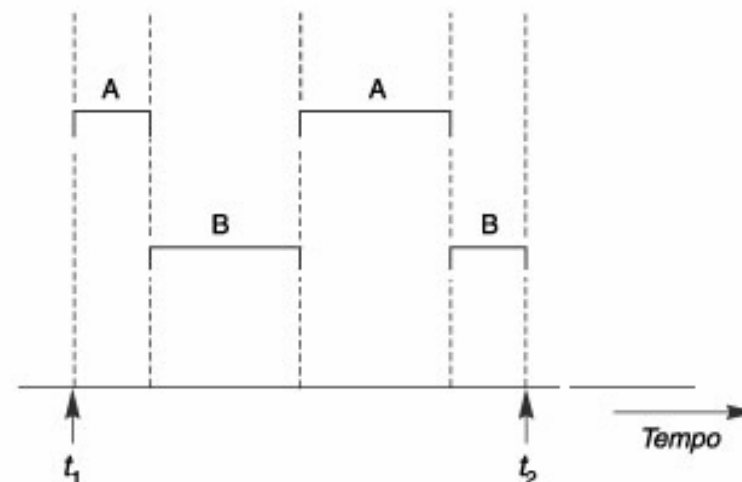
Necessidade do Controle de
Concorrência

SGBD Multiusuário

**Operações no BD:
Seriais**



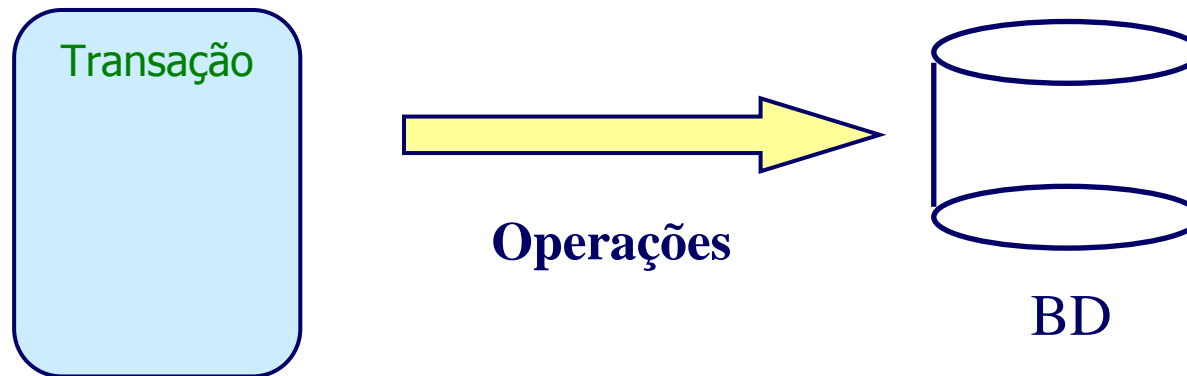
**Operações no BD:
Concorrência Intercalada**



Definição: Transação

Transação: programa em execução que forma uma **unidade lógica** de processamento no BD que deve ser **completo** e **integral**

Transação: inclui **uma ou mais operações de acesso ao BD** – inserção, exclusão, alterações ou consultas



Definição: Granularidade de Itens de Dados

Item de dados:

campo, registro, bloco do disco, arquivo, BD inteiro

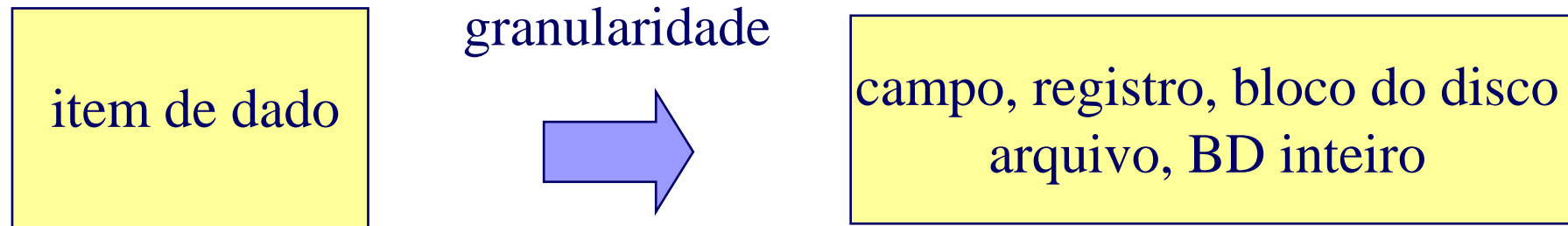
Granularidade = Tamanho dos itens de dados

Granularidade Fina: Itens de dados pequenos

Granularidade Grossa: Itens de dados grandes

Estrutura da Transação

Modelo simplificado: BD = coleção de itens de dados



T1: Transação T1

Início

- A=ler_item(X): lê um item de dado X do BD em uma variável A
- processa(A)
- escrever_item (X,A): escreve o valor da variável A em um item do BD X

Fim

Exemplos de Transações

T1: Transação T_1

```
A=ler_item (X);  
A = A - N;  
escrever_item (X,A);  
B=ler_item (Y);  
B = B + N;  
escrever_item (Y,B);
```

T2: Transação T_2

```
C=ler_item (X);  
C = C + M;  
escrever_item (X,C);
```

Transações de vários
usuários



- (1) executadas
concorrentemente
- (2) podem acessar e atualizar o
mesmo item de dados

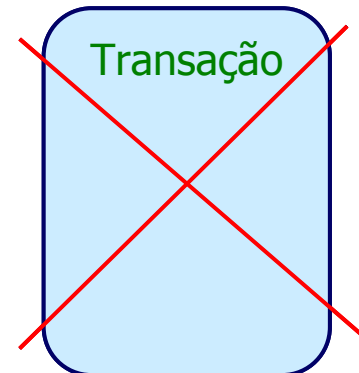
Transações do SGBD

Na execução de uma **transação** o **SGBD** deve garantir:

(1) **Todas as operações** na transação foram completadas com **sucesso** e seu efeito será **gravado permanentemente** no BD



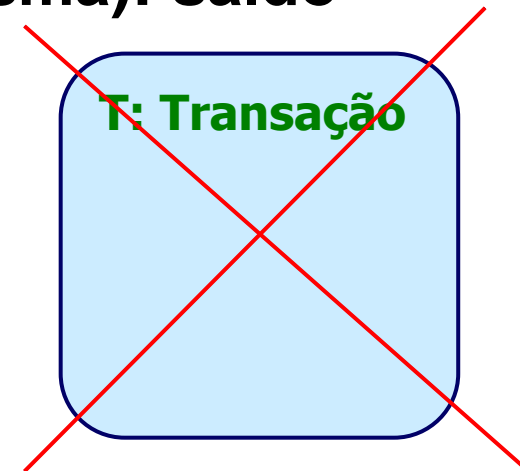
(2) Transação **NÃO** terá nenhum **efeito sobre o BD** ou outras transações (**TRANSAÇÃO FALHAR DURANTE EXECUÇÃO**)



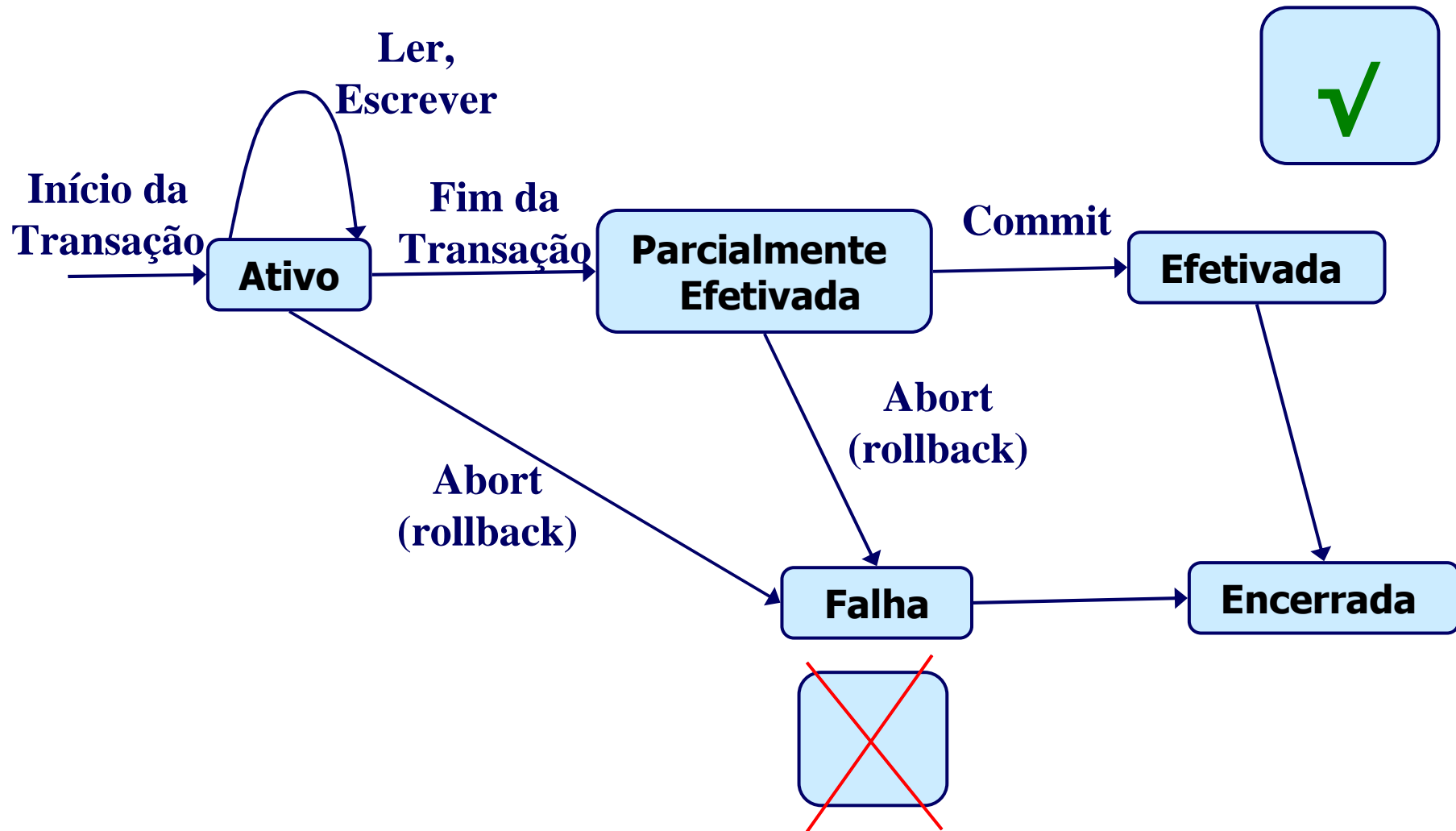
Tipos de Falhas

TIPOS DE FALHAS:

- ◆ Computador falhar por hardware, software ou rede
- ◆ Erro durante execução de operação na transação: estouro de variáveis
- ◆ Condições de exceção detectadas pela transação (necessitam o cancelamento da mesma): saldo insuficiente em conta
- ◆ Falta de energia, ar-condicionado



Estados de Execução de uma Transação



Propriedades da Transação

PROPRIEDADES

ACID



ATOMICIDADE
CONSISTÊNCIA
ISOLAMENTO
DURABILIDADE

Atomicidade

Transação = unidade atômica

(1) Executada em sua totalidade

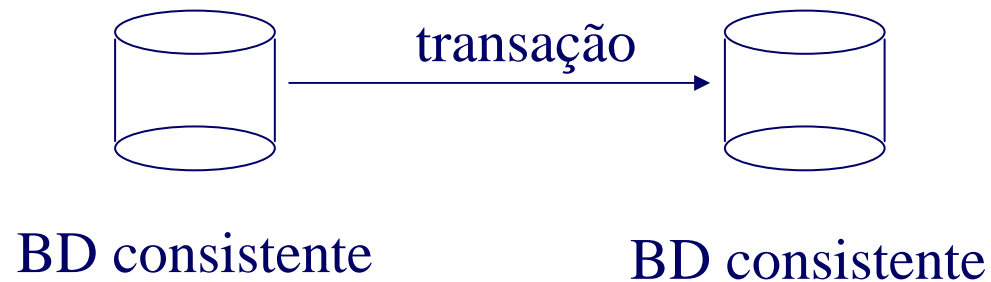
(2) Não pode executar somente partes  **operações desfeitas**



Consistência

Tarefa do programador

SGBD (restrições de integridade)



Isolamento

- ❑ A execução de uma **transação não pode ser afetada por outras** executando **concorrentemente** (o estado do sistema deve ser equivalente se transações são executadas uma de cada vez)
- ❑ Tudo deve se passar como se todos os recursos estivessem disponíveis

Durabilidade

- ❑ Os efeitos de uma transação confirmada (**commit**) não podem ser desfeitos

Planos de Execução das Transações

PLANOS de Execução: ordenação das operações de transação

T1: Transação T_1

```
A=ler_item (X);  
A = A - N;  
escrever_item (X,A);  
B=ler_item (Y);  
B = B + N;  
escrever_item (Y,B);
```

T2: Transação T_2

```
C=ler_item (X);  
C = C + M;  
escrever_item (X,C);
```

Planos Seriais

PLANOS SERIAS:

todas as operações das transações são executadas consecutivamente

Plano A

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item (X)

Plano B

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item(X,C);

Planos Seriais – Execução

Plano A

T1	T2
A=ler_item (X);	{A=90}
A = A - N;	{A=87}
escrever_item (X,A);	{X=87}
B=ler_item (Y);	{B=90}
B = B + N;	{B=93}
escrever_item (Y,B);	{Y=93}
{C=87}	C=ler_item (X);
{C=89}	C = C + M;
{X=89}	escrever_item (X)

Plano B

T1	T2
{C=90}	C=ler_item (X);
{C=92}	C = C + M;
{X=92}	escrever_item(X,C);
A=ler_item (X);	{A=92}
A = A - N;	{A=89}
escrever_item (X,A);	{X=89}
B=ler_item (Y);	{B=90}
B = B + N;	{B=93}
escrever_item (Y,B);	{Y=93}

X=90 Y=90 N = 3 M =2
X= 89 Y =93 ✓

Planos Não-Seriais

PLANOS NÃO-SERIAIS:

operações das transações executadas concorrentemente

Plano C

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item (X,C);

Plano D

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item(X,C);

Serialidade de Planos

SERIALIDADE DE PLANOS:

Identifica quais planos são corretos quando há intercalação das operações das transações na execução dos planos

PLANOS SERIALIZÁVEIS:

Equivalente a algum **plano serial** com as mesmas transações

Plano C

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item (X,C);

Plano D

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item(X,C);

Serialidade de Planos - Execução

Plano C

T1	T2
A=ler_item (X); A = A - N; {C=90} {C=92} escrever_item (X,A); B=ler_item (Y); {X=92} B = B + N; escrever_item (Y,B);	{A=90} {A=87} C=ler_item (X); C = C + M; {X=87,oper.perdida} {B=90} escrever_item (X,C); {B=93} {Y=93}

X=90 Y=90 N = 3 M =2
X= 92 Y =93 X

Plano não-serial C ->
Não é serializável

Plano D

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); {C=87} {C=89} {X=89} B=ler_item (Y); B = B + N; escrever_item (Y,B);	{A=90} {A=87} {X=87} C=ler_item (X); C = C + M; escrever_item(X,C); {B=90} {B=93} {Y=93}

X=90 Y=90 N = 3 M =2
X= 89 Y =93 ✓

Plano não-serial D ->
É serializável

Escalonamento de Planos

Escalonamentos ou planos de Execução

Plano A

T1	T2
A=ler_item (X); A = A - N; escrever_item (X,A); B=ler_item (Y); B = B + N; escrever_item (Y,B);	C=ler_item (X); C = C + M; escrever_item (X)

Plano A: EA=r1[x];w1[x];r1[y];w1[y];c1;r2[x];w2[x];c2;

Escalonamento Equivalentes

Plano A: $EA = r1[x]; w1[x]; r1[y]; w1[y]; c1; r2[x]; w2[x]; c2;$

Plano B: $EB = r2[x]; w2[x]; c2; r1[x]; w1[x]; r1[y]; w1[y]; c1;$

Plano C: $EC = r1[x]; r2[x]; w1[x]; r1[y]; w2[x]; c2; w1[y]; c1;$

Plano D: $ED = r1[x]; w1[x]; r2[x]; w2[x]; c2; r1[y]; w1[y]; c1;$

Plano X: $EX = r1[x]; w1[x]; r2[x]; r1[y]; w2[x]; c2; w1[y]; c1;$

$EA \equiv EB \equiv ED \equiv EX$

Escalonamentos corretos: escalonamentos seriáveis

- ❑ Um **escalonamento** é **seriável** se for **equivalente** a algum **escalonamento serial**.
- ❑ Dado E (**seriável**), podem existir mais de um escalonamentos seriais equivalentes.
- ❑ Escalonamentos **não-seriáveis** devem ser **proibidos** pelo Escalonador.

$EA \equiv EB \equiv ED \equiv EX$

Seriais: $EA \equiv EB$

Seriáveis: $ED \equiv EX$

Não-seriável: EC

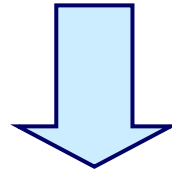
Técnicas para Controle de Concorrência

TCC: Assegurar a **serialização** de planos de execução usando **PROTÓCOLOS** (conjunto de regras):

- **Técnica de Bloqueio dos Itens de Dados**
(usado na maioria dos SGBDs comerciais)

Técnicas para Bloqueio de Itens de Dados

Técnica de bloqueio dos itens de dados: impedir que múltiplas transações acessem os itens concorrentemente



SE a transação precisar de um **item bloqueado**
ENTÃO forçada a **esperar** até que o item seja **liberado**

Técnica de Bloqueio de Itens de Dados

Funcionamento

Bloqueio (lock)
do item de dados



variável associada ao
item de dados

Bloqueio para
CADA item de dados



sincronizar o acesso por
transações concorrentes
aos itens do BD

Bloqueio Binário

LOCK(X) = 2 estados, X – item de dado

Dois Estados: (1) bloqueado (0) desbloqueado

- Se o **valor do bloqueio em X** for **1** o item X não pode ser acessado por uma operação de BD, se for **0** pode ser acessado
- Um **bloqueio distinto** é associado a cada item X do BD
- Operações: **lock_item(X)** e **unlock_item(X)**

Bloqueio Binário - Exemplo

Bloqueio Binário para as Transações T_1 e T_2

T1: Transação T_1

lock_item(X);

A=ler_item (X);

A = A - N;

escrever_item (X,A);

unlock_item(X);

lock_item(Y);

B=ler_item (Y);

B = B + N;

escrever_item (Y,B);

unlock_item(Y);

T2: Transação T_2

lock_item(X);

C=ler_item (X);

C = C + M;

escrever_item (X,C);

unlock_item(X);

Bloqueios Compartilhados/Exclusivos

Bloqueios Compartilhados/Exclusivos ou de Leitura/Escrita:

Bloqueio Compartilhado:

Diversas transações podem acessar o mesmo item X se todas foram com o propósito de **Leitura**.

Bloqueio Exclusivo:

Se uma transação for alterar (**Escriver**) um item X

Bloqueios Compartilhados/Exclusivos - Estados

LOCK(*X*) = 3 estados, *X* – item de dado

Três estados:

read_locked – **bloqueio compartilhado**

permite que outras operações leiam o item

write_locked – **bloqueio exclusivo**, uma

uma única transação controla

exclusivamente o bloqueio no item

unlocked - **item desbloqueado**

Operações:

read_lock_item(*X*), **write_lock_item(*X*)** e **unlock_item(*X*)**

Bloqueio Compartilhado/Exclusivo - Exemplo

Bloqueio Compartilhado/Exclusivo para as Transações T_1 e T_2

T1: Transação T_1
write_lock_item(X);
A=ler_item (X);
A = A - N;
escrever_item (X,A);
unlock_item(X);
write_lock_item(Y);
B=ler_item (Y);
B = B + N;
escrever_item (Y,B);
unlock_item(Y);

T2: Transação T_2
write_lock_item(X);
C=ler_item (X);
C = C + M;
escrever_item (X,C);
unlock_item(X);

Bloqueio Compartilhado/Exclusivo x Binário

Transação T_3

T3: Transação T_3
A=ler_item (X);
B=ler_item (Y);
imprimir(A + B);

Bloqueio Binário

T3: Transação T_3
lock_item(X);
A=ler_item (X);
unlock_item(X);
lock_item(Y);
B=ler_item (Y);
unlock_item(Y);
imprimir(A + B);

Bloqueio Compartilhado/Exclusivo

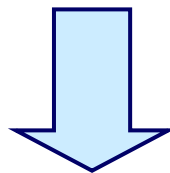
T3: Transação T_3
read_lock_item(X);
A=ler_item (X);
unlock_item(X);
read_lock_item(Y);
B=ler_item (Y);
unlock_item(Y);
imprimir(A + B);

O Bloqueio Compartilhado/Exclusivo é menos restritivo que o Bloqueio Binário.

Granularidade de Itens de Dados

Técnicas de CC assume o BD = conjunto de itens de dados:

- **Campo**
- **Registro**
- **Bloco do Disco**
- **Arquivo**
- **BD inteiro**

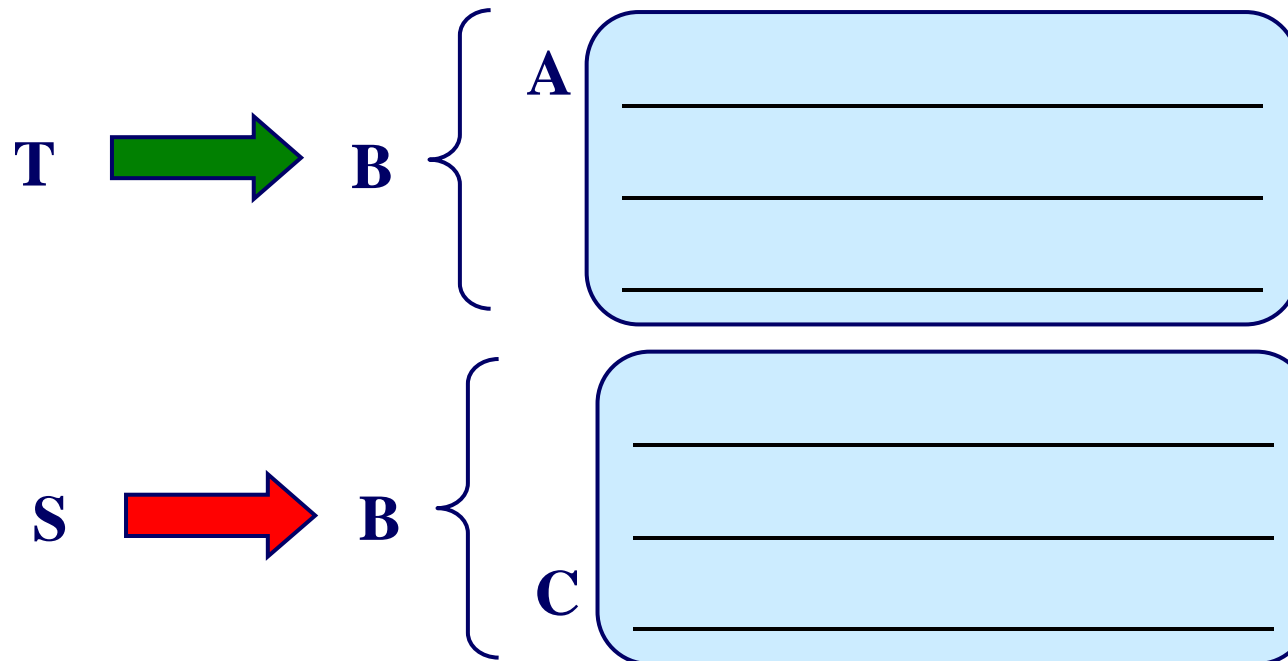


A escolha da Granularidade (grossa ou fina) pode afetar a execução do controle de concorrência

Granularidade - Exemplo

Tamanho do item de dados = bloco do disco

- (a)** Se uma transação T precisa bloquear um registro A deve bloquear todo o bloco B que contém A
- (b)** Outra transação S quer bloquear um registro C armazenado no bloco B é forçada a esperar

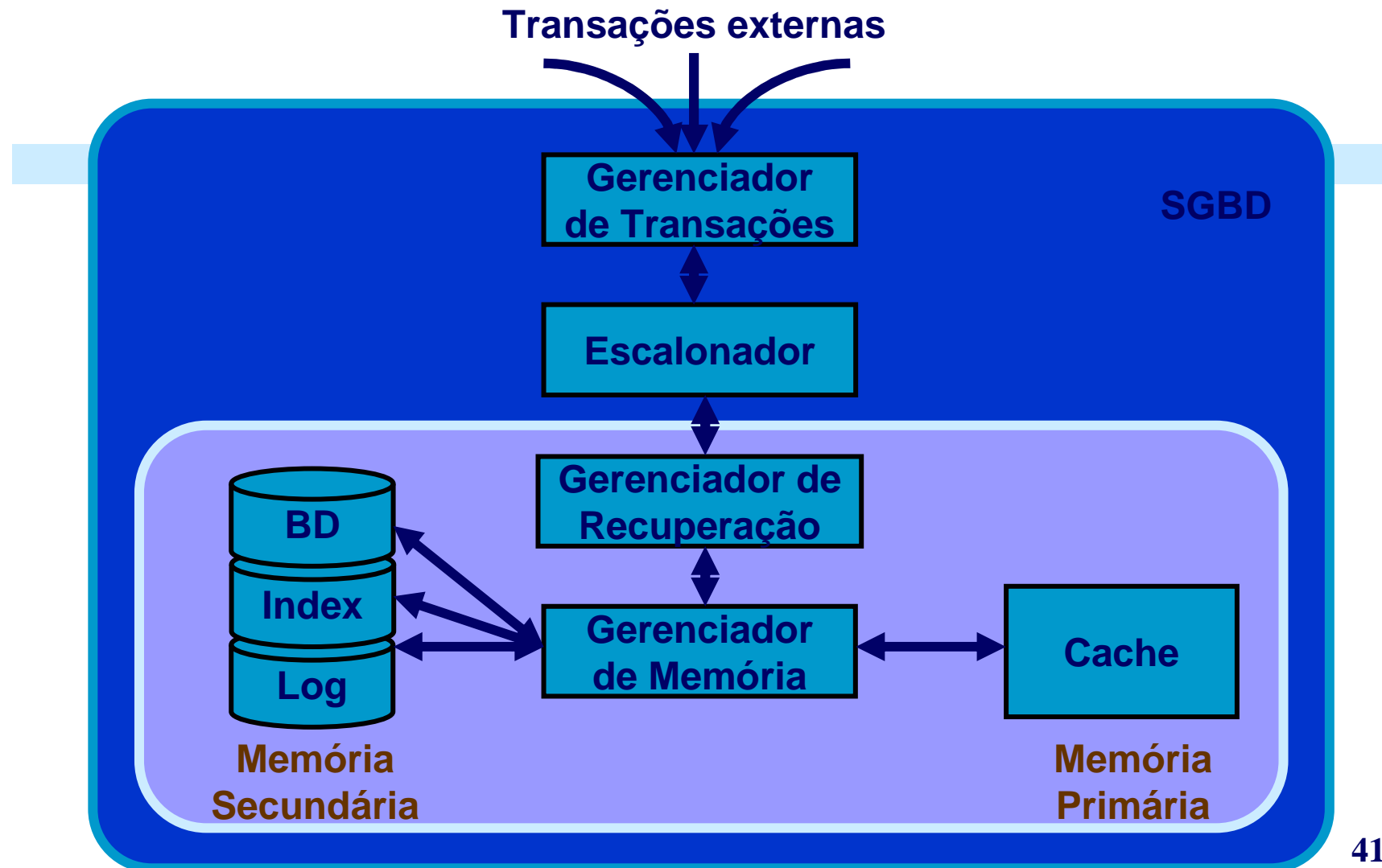


Granularidade de Itens de Dados

Quanto maior o tamanho do item de dados mais baixo é o grau de concorrência permitido



Gerenciador de Memória e Recuperação



SUMÁRIO

SGBD multiusuário (centralizado ou distribuído) necessita do CC

Transação: programa composto por operações no BD

Propriedades das Transações: ACID

Planos de Execução: Seriais, Não-Seriais, Seriáveis

Plano Seriável: equivalente a alguma **plano serial** com as mesmas transações

Técnicas para CC: bloqueio dos itens de dados

CC: fundamental para permitir o uso mais **eficiente** dos **recursos computacionais**

BIBLIOGRAFIA

- **Elmasri, R.; Navathe, S. B. Sistemas de Banco de Dados, 4ed. Pearson Addison Wesley, 2005 .724 p.**
- **Silberschatz, A.; Korth, H. F., Sudarshan, S. Sistema de Bancos de Dados. 3a. Edição, Makron Books, 1998. 778p.**
- **Ferreira, J.E.; Finger, M., Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas, Coleção de textos especialmente preparada para a Escola de Computação, 12a, São Paulo, 2000. Cópia em: <http://www.ime.usp.br/~jef/ec2000.ps>**
- **Date, C. J. Introdução a Sistemas de Banco de Dados. Ed. Campus, 1990.**